

REMARKS

Claims 1-33 are pending in this application. Claims 32 and 33 have been amended to correct minor informalities.

Applicants wish to thank the Examiner for the indication that claims 16, 17 and 33 recite allowable subject matter.

Claims 1, 2, 5, 9, 10, 13, 19, 20, 23, 27 and 32 stand rejected under 35 U.S.C. § 102(b) as being anticipated by U.S. Patent No. 5,313,614 (hereinafter "Goettelmann"). Claims 3, 4-8, 11, 12, 14-18, 21, 22, 24-26 and 28-31 stand rejected under 35 U.S.C. § 103(a) as being unpatentable over Goettelmann in view of U.S. Patent No. 6,021,275 (hereinafter "Horwat"). Reconsideration of these grounds of rejection is respectfully requested for the following reasons.

The undersigned spoke with the Examiner on May 24, 2005 to suggest that a telephonic interview may be helpful in resolving the issues in this case. The Examiner felt that it would be better for the Applicants to respond on the merits and then to hold an interview at a later date if necessary. The undersigned would be grateful if the Examiner would contact him by telephone if after considering the responses presented below, the Examiner believes that the responses do not resolve the issues in this case.

I. Claim 32 Incorporates The Features Of Allowed Claim 16 And Should Therefore Also Be Allowed

Independent claim 32 incorporates the same features as recited in allowed dependent claim 16, namely that "the code translator runs as a user mode process under control of a host operating system on the host computer, and wherein the interpreter runs as a kernel mode driver thread under the host operating system." Because the Office Action indicates that the subject matter of claim 16 is allowable, claim 32 should be allowed for the same reasons. Reconsideration of the rejection of claim 32 is respectfully requested on this basis.

II. Goettelmann Does Not Anticipate The Subject Matter of Claims 1, 2, 5, 9, 10, 13, 19, 20, 23 & 27

As Applicants have explained in prior responses, the present invention takes a novel approach to emulating the execution of a computer program comprising instructions of the

instruction set of a target (first) computer on a host (second) computer having a different instruction set. This novel approach involves two steps. First, the instructions of the target program are statically translated into a series of instructions of an intermediate instruction set. The intermediate instruction set is optimized for interpretation on the second computer. The intermediate instruction set is **not** the native instruction set of either the first computer or the second computer. Second, the series of instructions of the intermediate instruction set are then **directly** interpreted on the second computer to emulate execution of the program; no further translation into the native assembly code or machine (object) code of the second computer is performed. These features are recited in each of the independent claims 1, 9, 19, and 27.

Goettelmann does not teach **directly** interpreting the instructions of an **intermediate** instruction set on the second computer, and therefore, Goettelmann does not anticipate independent claims 1, 9, 19 and 27. On the contrary, Goettelmann expressly describes **further translating** an intermediate set of instructions into the native assembly language of the second computer. Specifically, the translation process of Goettelmann is described at column 15, line 58 to column 16, line 2, as follows:

The translation process is similar in many respects to a conventional compiler process and, as such, has three principal components: parsing, transformation and code generation. In particular, the parsing component reads the input application software or system software object code and converts it into an *intermediate language representation* suitable for further, machine-independent processing. The transformation component then reduces the size of the intermediate language code by eliminating redundancies and unnecessary code. *Finally, the code generation component converts the intermediate language representation to the object language of the target machine.*

(emphasis added). As the highlighted sentence at the end of this portion expressly states, the process of Goettelmann involves **further converting** the intermediate language representation to the native instruction set (“object language”) of the second computer. Goettelmann provides additional description of this final code generation step later in the specification:

The code generation phase of the translation process is virtually the same as the code generation phase of a conventional compiler. *That is, the intermediate language version of the*

application, optimized as described above, is converted to the assembly language of the target machine by substituting one or more appropriate target machine assembly language instructions for each intermediate language instruction. The resulting code is thereupon assembled into target machine object code and bound with the translated system software.

(Col. 29, ll. 56-65) (emphasis added). Here, again, Goettelmann clearly states that the intermediate language representation is *further converted* to the native assembly language of the second computer. In the preferred embodiment, as mentioned in the last sentence of this quote, the “resulting [assembly] code” can then be “assembled into target machine object code.” This is directly contrary to the claimed invention, in which the intermediate representation is itself *directly* interpreted (not compiled) on the second computer.

In asserting that Goettelmann does teach directly interpreting the intermediate representation, the Office Action misreads a statement at the end of Goettelmann’s specification. Specifically, Goettelmann states:

Moreover, it may be desired to execute the translated program in assembly form (using an interpreter) rather than compiling it into target machine object form.

All that Goettelmann is saying here is that after *further converting* the intermediate representation into the native assembly code of the second computer (as discussed above), that resulting assembly code may be executed by interpretation instead of compiling the resulting assembly code into object form. But, significantly, it is *not* the instructions of the intermediate representation that Goettelmann says can be interpreted, but rather the assembly code that results from *further converting* the intermediate representation into that assembly form. Goettelmann is in no way stating, nor even remotely suggesting, that the instructions of the intermediate representation be directly interpreted on the second computer. Consequently, Goettelmann does not anticipate the subject matter of independent claims 1, 9, 19 and 27, and the Section 102(b) rejection of these claims should be withdrawn.

Because dependent claims 2, 5, 10, 13, 20 and 23 each depend either directly or indirectly from one of independent claims 1, 9, 19 and 27, Applicants submit that they too are not anticipated by Goettelmann for the same reason. Nevertheless, Applicants wish to point out that there are additional reasons why these dependent claims are not anticipated by Goettelmann.

Claim 2 recites that “the intermediate instruction set comprises a plurality of control words that are derived, at least in part, by mapping control words of the instruction set of the target machine into the fundamental word size of the host machine.” Claim 27 recites a similar feature. The specification explains that this feature facilitates high speed interpretation of the intermediate instruction set on the second (host) computer:

In order to facilitate high speed interpretation on the host machine in accordance with the present invention, all operands and control words of the intermediate instruction set are mapped into 64 bits, the fundamental word size of the host machine (e.g., a 64-bit Alpha 20164).

Spec., p. 10, ll. 17-19. The Office Action asserts that this feature is disclosed in Goettelmann at column 20, lines 5-15. Applicants do not understand how the cited portion of Goettelmann teaches the claimed feature of deriving an intermediate instruction set by mapping control words of the instruction set of the first computer into the fundamental word size of the host machine. The cited portion of Goettelmann does not appear to be discussing such a concept at all. Reconsideration is respectfully requested. If the Examiner persists in the rejection of this claim, Applicants respectfully request that the Examiner provide a more detailed explanation for how the cited portion of Goettelmann teaches the claimed subject matter.

Claim 5 recites that the control words of the intermediate instruction set are “derived from control words of the instruction set of the target machine in a manner that reduces the number of different forms of control words in the intermediate instruction set.” This results in a reduction in the complexity of the intermediate instruction set, and hence better performance during interpretation. *See*, Specification, p. 11, ln. 22 – p. 12, ln. 12. The Office Action asserts that this feature of the invention is disclosed in Goettelmann at column 15, line 65 to column 16, line 5. That is incorrect. The cited portion of Goettelmann merely states that after a program has been translated into the intermediate language representation, the size of the intermediate language code *of the translated program* is reduced by eliminating redundancies and unnecessary code in the translated program. Claim 5 is instead directed to how you create and optimize the instruction set of the intermediate language itself. The cited portion of Goettelmann has nothing to do with creating the instruction set of an intermediate language. Reconsideration of the rejection of claim 5 is therefore respectfully requested.

Inasmuch as the remaining dependent claims 10, 13, 20 and 23 recite features similar to those discussed above, Applicants submit that they are not anticipated by Goettelmann for the same reasons.

III. Horwat Does Not Teach The Features of Dependent Claims 3, 4-8, 11, 12, 14, 15, 21, 22, 24-26 and 28-31

While claims 3, 4-8, 11, 12, 14, 15, 21, 22, 24-26 and 28-31 also depend either directly or indirectly from one of independent claims 1, 9, 19 and 27 and are therefore patentable for the same reasons as discussed above, Applicants nevertheless submit that the Office Action is incorrect in asserting that the subject matter of any of these claims is taught by Horwat. Like the dependent claims discussed above, these claims also recite features of the intermediate instruction set that result in optimized interpretation of those instructions on the second computer. Horwat does not teach the features recited in these claims.

Claim 3 recites that some of the control words of the intermediate instruction set are “derived by redefining control words of the target computer to minimize the number of masking and shifting operations needed to decode the plurality of control words of the intermediate instruction set.” The Office Action asserts that Horwat teaches this feature at column 21, lines 62 to column 22, line 15 and also lines 20-25 of its specification. However, this portion of Horwat does not describe how to define or create an intermediate instruction set itself, but instead describes how to translate a data structure during a code translation operation. Moreover, there is no mention in the cited portion of Horwat of “redefining control words of [a] target computer *to minimize the number of masking and shifting operations* needed to decode [a] plurality of control words of the intermediate instruction set.” Reconsideration of the rejection of claim 3 is respectfully requested. If the Examiner persists in the rejection of this claim, Applicants respectfully request that the Examiner provide a more detailed explanation for how the cited portion of Horwat teaches the claimed subject matter.

Claim 4 recites that the control words of the intermediate instruction set are defined “to minimize the time needed to determine the type of a control word.” The Office Action asserts that Horwat teaches this feature at column 20, lines 45-63. This is incorrect. The cited portion of Horwat describes the format of certain directives that are included in a “data translation script” that is used by a translator in “converting literal data represented in big-

endian format into little-endian format.” Col. 20, ll. 35-43. The cited portion of Horwat does not describe how to define or create an intermediate instruction set, let alone defining control words of such an intermediate instruction set to minimize the time needed to determine the type of such control words during interpretation. Reconsideration of the rejection of the claim 4 is respectfully requested for this reason.

The Office Action does not appear to address dependent claim 6. If the Examiner persists in the rejection of this claim, Applicants respectfully request that the Examiner provide a detailed explanation for the rejection.

Claim 7 recites that “the instructions of the intermediate instruction set have a fixed length and do not cross code word boundaries.” The Office Action asserts that Horwat teaches this feature at column 24, lines 63-65. On the contrary, these lines of Horwat simply provide a list of certain “operand encodings” for “Register numbers” and “Integers.” There does not appear to be any mention of the length of any such operands, nor whether they are defined to cross code word boundaries or not. Reconsideration of the rejection of claim 7 is therefore respectfully requested. If the Examiner persists in the rejection of this claim, Applicants respectfully request that the Examiner provide a more detailed explanation for how the cited portion of Horwat teaches the claimed subject matter.

Claim 8 recites that “zero-address instructions of the instruction set of the *target* [*i.e.*, first] machine for pushing data onto a stack for use in a subsequent zero-address instruction operation *are incorporated as explicit addresses into a new instruction in the intermediate instruction set for performing that operation*, thereby reducing the number of different instructions in the intermediate instruction set.” The Office Action asserts that this feature is disclosed in the table that appears at the bottom of column 30 of Horwat. However, the specification of Horwat explains that the table at the bottom of column 30 is not a table at all, but rather a listing of the code that a “batch translator” would produce and execute when passed an object code file called “hello.o”. Col. 30, ll. 49-51 (“A batch translator implementation of Intercode translator 220c produces and executes the following code when passed hello.o.”). Applicants cannot understand how this code listing could possibly teach or suggest the claimed feature of incorporating zero-address instructions of the instruction set of a target machine for pushing data onto a stack as explicit addresses into a new instruction in an intermediate instruction set for performing that operation.” Reconsideration of the

DOCKET NO.: USYS-0111/TN137
Application No.: 09/294,617
Office Action Dated: February 25, 2005

PATENT

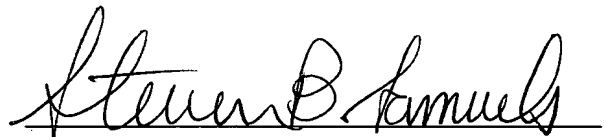
rejection of claim 8 is therefore respectfully requested. If the Examiner persists in the rejection of this claim, Applicants respectfully request that the Examiner provide a more detailed explanation for how the code listing at the bottom of column 30 of Horwat teaches the claimed subject matter.

The Office Action rejects the remaining dependent claims 12, 14, 15, 18, 20, 24-26 and 28-31 on the same basis as claims 3, 4, 7 and 8. The Office Action does not provide any reasons for rejecting claims 21 and 22. To the extent that these remaining claims incorporate features similar to those recited in claims 3, 4, 7 and 8, Applicants respectfully submit that they patentably define over Horwat and the other cited art of record for the same reasons as discussed above.

CONCLUSION

For all the foregoing reasons, Applicants respectfully submit that the present application is in condition for allowance. An early notice of allowance is respectfully requested.

Date: *May 25, 2005*


Steven B. Samuels
Registration No. 37,711

Woodcock Washburn LLP
One Liberty Place - 46th Floor
Philadelphia PA 19103
Telephone: (215) 568-3100
Facsimile: (215) 568-3439

CORRESPONDENCE ADDRESS:

Michael B. Atlass
Registration No. 30,606
UNISYS CORPORATION
Unisys Way, MS/E8-114
Blue Bell, Pennsylvania 19424